Parallel computing for eddy current testing simulation

Laurent Santandrea, Guilhem Savel, Yann Le Bihan, Adel Razek

Parallel computing for eddy current testing simulation

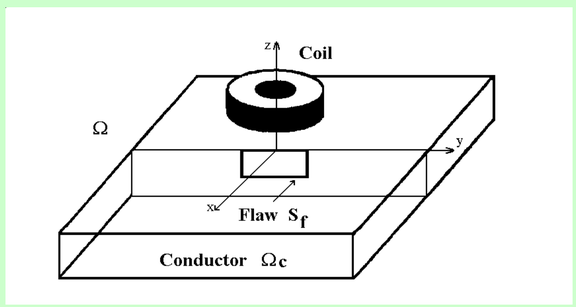
Laurent SANTANDREA, Guilhem SAVEL, Yann LE BIHAN, Adel RAZEK
Laboratoire de Génie Électrique de Paris / SPEE Labs, CNRS UMR 8507, Universités ParisVI et Paris XI, Supelec, Plateau de Moulon 91192 - Gif sur Yvette - France
E-mail: santandrea@lgep.supelec.fr

Introduction

Eddy current testing (ECT) numerical simulation may need excessive computational time; e.g. a 3D modelling to detect a thin flaw with an air-coil probe. The computational time along a scan line can reach half a day on an Intel Pentium 4 3.2 GHz processor. It is obvious that computational effort becomes prohibitive if we want to optimise the probe or to reconstruct the flaw (inverse problem).

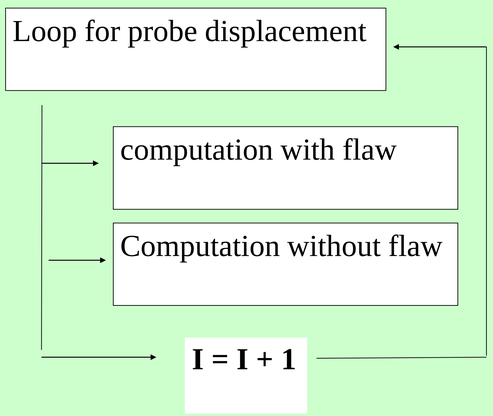
Dual-processors with hyper-threading technology is became a current architecture, and can be obtained with a very reasonable cost. We will see that OpenMP is a well adapted solution to parallelize easily a code and to exploit this machine architecture.

Eddy Current Testing (ECT) Problem



ECT-NDT configuration

Excessive computational time



Scheme of a ECT modelling problem

Ref ISEM

Parallel Computing

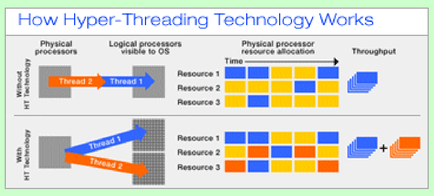
The computational technology evolution has involved radical modifications in the applications development.

In particular the increase of performing computer park as consequence the decrease of cost and the new powerful processors appearance.

The fast Ethernet network with large bande has allowed the computer network group.

MPI is a library, which is used on distributed memory systems. For MPI the management of these communications is of the user responsibility. MPI is thus often more delicate to be implemented.

OpenMP is based on compiler directives, and is generally used on shared memory machines. The great advantage of OpenMP is that the communication mode between the threads is managed directly by the compiler. In particulary, OpenMP is well adapted to hyper-threading technology. This technology is developed by Intel corporation allows a single processor to manage data as if it was two processors by executing data instructions from different threads in parallel rather than serially.



IMPLEMENTION

A computer dual-XHEON with two micro-processors Intel Pentium 4 with hyper-threading technology is used on a LINUX operating system (distribution Mandrake 10.1).

Define number of CPU :

terminal>> export OMP_NUM_THREADS=4;

Compile our program :

terminal>> ICC –openmp monprog.cpp

Speedup = execution time with N CPU / execution time with one CPU

	Sequential	Parallel
CPU time	1208.7 s	333.34 s

Speedup : 3.7

```
#include <omp.h>

.....
.

for (i=0; i<N;i=i+2) {
 #pragma omp parallel sections
 {
   #pragma omp  section
    {
     AV[i].fem("msh3d","phys");
    }
   #pragma omp  section
    {

AVf[i].fem("msh3d","phys","fis");
   }
   #pragma omp  section
      {
       AV[i+1].fem("msh3d","phys");
      }
     #pragma omp  section
      {

AVf[i+1].fem("msh3d","phys","fis")
;
      }
  }
 } // fin de boucle de position
```

Conclusion

Computational time saving is quasi-proportional to processors number and can be better with a true multi-processors architecture. The proposed method can be used of course with a single hyper-threaded processor such as Pentium III and Pentium 4 processors to exploit parallelism. This technique is well adapted to the ECT problem which is easily divisible in independent problems.